深度强化学习复原多目标航迹的 TOC 奖励函数 *

贺　亮[1]，徐正国[1]，贾　愚[1]，沈　超[2]，李　赟[1]

(1. 盲信号处理重点实验室，成都 610041; 2. 西安交通大学 智能网络与网络安全教育部重点实验室，西安 710049)

摘　要：针对航迹探测领域中探测器获得的目标地理位置通常是同一帧下无法区分的多目标场景，需要利用目标位置信息还原各航迹并区分各目标的问题进行研究，提出采用深度强化学习方法复原目标航迹的方法。依据目标航迹的物理特点，提取数学模型，结合目标航迹的方向、曲率等提出轨迹曲率圆(trajectory osculating circle，TOC)奖励函数，使深度强化学习能够有效复原多目标航迹并区分各目标。首先描述多目标航迹复原问题，并将问题建模成深度强化学习能够处理的模型;结合 TOC 奖励函数对多目标航迹复原问题进行实验;最后给出该奖励函数的数学推导和物理解释。实验结果表明，TOC 奖励函数驱动下的深度强化网络能够有效还原目标的航迹，在航向和航速方面切合实际目标航迹。

关键词：深度强化学习；序贯决策；Q 函数；轨迹密切圆

中图分类号：TP391　　doi: 10.19734/j.issn.1001-3695.2018.12.0886

Design of reward function in multi-target trajectory recovery with deep reinforcement learning

He Liang[1], Xu Zhengguo[1], Jia Yu[1], Shen Chao[2], Li Yun[1]

(1. *National Key Laboratory of Science & Technology on Blind Signal Processing*, *Chengdu* 610041, *China*; 2. *MOE Key Laboratory for Intelligent Networks & Network Security*, *Xi'an Jiaotong University*, *Xi'an* 710049, *China*)

Abstract: It attracts lots of attention in the field of object trajectory detection that detectors always receive several geographical locations without any other information about the targets, and furthermore it comes into a problem to use the geographical location information received by the sensors to reconstruct the trajectories of each target as well as to distinguish the targets in each frame, which is called multi-target trajectory recovery and can be solved by Deep Reinforcement Learning (DRL) . A trajectory osculating circle (TOC) reward function is implemented based on the mathematical model of the direction and trajectory curvature according to the peculiarity of trajectories in actual. Firstly, the issue of the multi-target trajectory reconstruction is switched into a model which can be appropriate for DRL. Then, DRL is tested with the proposed reward function. Finally, a mathematical derivation and physical interpretation of the proposed TOC reward function is introduced. The experimental result shows that with the guidance of the TOC reward function, DRL can reverse the trajectory effectively, and the trace corresponds well with the actual trajectory.

Key words: deep reinforcement learning; sequential decision; q-function; trajectory osculating circle

0　引言

强化学习(reinforcement learning)在具有决策性质的问题中逐渐凸显出了优异的性能而备受关注。强化学习的理论基础由文献[1]提出，该方法模拟了生物根据环境的影响来自动调节自身的行动以最好的适应环境。Mnih 等人[2]通过强化学习从高维输入信息中利用深度学习模型成功学习到了控制策略，并在 Atari 游戏上取得成功。为了模拟生物对环境的适应性，需要解决如下问题：a)从高维的输入信息中找出有效的环境信息，如从视觉图片中找到游戏中的攻击目标等；b)根据环境信息作出有效的决策以改变环境以达到有利的状态及最终结果[3,4]。通过对 Atari 的图像像素信息和游戏得分作为输入，根据游戏环境采用强化学习方法训练出的决策可以与职业游戏玩家的技能相当。可以看出，对于动态决策问题，为了能够得到最优的最终结果，每一步决策未必要选择当前一步最优的，而强化学习可以通过决策的学习过程逐步学习到如何使最终结果达到最优的决策方法。

在深度学习被提出之前，局限于内存复杂度、计算复杂度以及机器学习算法和采样复杂度等问题，强化学习的稳定性较差，并且只局限于解决低维输入问题[5,6]。深度学习与传统的神经网络相比有更多的隐层，从而具有更多的超参数[7]，激活函数从 Sigmoid 改变成 ReLU[8]，深度神经网络具有很强函数逼近能力和学习能力等特点，从而为强化学习解决高维复杂问题提供了有力的工具。结合了深度学习的强化学习又被称为深度强化学习(deep reinforcement learning, DRL)[9]。然而，二者的简单结合并不能保证学习过程的稳定性，因此有一系列致力于稳定性的研究，目前主要的方法有:a)通过引入重放(replay)的机制，让智能体在训练的过程中定期重复之前玩过的游戏从而加强对类似环境问题的解决策略的学习[10~12]，然而，该过程会消耗较多内存用于存储历史游戏信息;b)利用多个智能体并行训练，并解耦各个并行智能体之间的数据，使得每个智能体处理的环境信息具有稳定性[13]，同时还

收稿日期: 2018-12-27; **修回日期:** 2019-01-19　**基金项目:** 国家自然科学基金重点项目（面向大规模多源数据的人物画像及定位技术，U1736205）；国家自然科学基金项目（触感行为为特征识别的移动智能终端隐式身份认证方法研究，61773310）

作者简介: 贺亮（1990-），男，黑龙江佳木斯人，博士，工程师，主要研究方向为人工智能、网络协议分析（lianghe@sei.xjtu.edu.cn）；贾愚（1990-），男，博士，工程师，主要研究方向为智能信息处理；沈超（1984-），男，博士，副教授，主要研究方向为智能信息处理、智能穿戴设备行为分析；李赟（1984-），男，博士，工程师，主要研究方向为网络态势感知.

可以利用 GPU 进行加速[14,15]或采用深度强化学习的分布式架构[16]。

深度强化学习能够处理强化学习不能处理的更复杂问题，在内存空间增大、并行技术发展的情况下，稳定性也得到较好的解决。利用深度神经网络处理 Atari 游戏视觉图像的高维信息，提取出的关键信息使得强化学习能够更加稳定的处理该游戏中的决策问题。AlphaGo 能够成功击败人类，也是利用深度强化学习以及其他的经典搜索算法[17]。还有许多其他复杂难操作的游戏均可由深度强化学习来完成，如 FlappyBird[18,19], TORCS[20]等。这些实验均在 OpenAI Gym 平台[21]上实验。

目标航迹聚类问题[22]主要针对海量的航迹数据中挖掘各个目标的航行轨迹，根据目标的属性信息进行聚类，一般是根据目标的航距航速等信息综合判断各个点属于各个目标的聚类结果。该问题一般可以采用 K-means 等聚类方法针对目标的空间地理位置信息和身份信息进行聚类[23]。然而 K-means 等传统聚类方法由于聚类过程中重点针对距离信息进行聚类计算[24]，引入其他属性需要设计更加复杂的度量距离的函数，或者将输入数据通过核函数映射到相应的高维空间，但是核函数的设计不具有直观性。本文利用深度强化学习中奖励函数的设计，直观引入航迹的几何特性曲线的数学特性，所设计的奖励函数更加直观有效，避免了聚类算法中核函数的复杂且不直观的设计工作。

深度强化学习使计算机能够较好的处理复杂的决策类游戏，抛开游戏性不谈，航迹复原问题也可以交给深度强化学习来解决。本文所要解决的航迹复原问题如下：探测器按照一定频率获得多个目标的地理位置信息，但无法区分各个目标，期望以此为各个目标画出航迹，由于探测器的限制，存在对目标的漏检和虚报。该问题可以刻画为需要在各个时刻进行决策如何连线的动态规划问题。本文采用深度强化学习方法对该问题进行解决，并提出一种适用于该问题的奖励函数。

1　序贯决策问题与强化学习

强化学习是通过不断对环境重复实验，从中积累一定的经验，从而能够实现智能决策的学习过程。强化学习中包含智能体(agent)，环境(environment)和奖励(reward)，在强化学习过程中，智能体和环境通过奖励机制完成一定的交互。在每一时刻环境都处于某种状态(state)，智能体根据对环境状态的观测值，依据一定的策略作出行动(action)，该行动会影响环境，从而使环境进入下一个状态，智能体再从改变了状态后的环境中获得新的环境状态以及它采取了行动后环境反馈的奖励。强化学习中涉及的针对不同时刻下环境的状态采取不同的策略，即为典型的序贯决策问题，同时策略和环境之间的数学描述需要采用状态值函数以及状态-行动值函数来描述，为了得到全局最优解，还需要在决策过程中进行探索与开发。

1.1　序贯决策

序贯决策是依时间顺序，在各个时刻点上根据环境的状态情况进行的决策。对各个阶段的决策形成策略。在强化学习过程中，根据有回报的环境行动交互数据，每个阶段下有利于实现目标的动作被保留，不利于实现目标的动作被抛弃。

1.2　状态值函数

策略是选择动作的依据，在采用某种策略后，得到对应奖励的期望值即为状态值函数。强化学习的过程就是需要根据以往的实验结果针对各个策略构建一个指标函数，从而依据该指标渐进收敛到最优策略。例如，贪婪策略就是在每一步决策过程中选择当前能够获得的状态值最大的那个策略，然而依此法选择的策略未必会是全局最优策略。

1.3　状态-行动值函数

在采取某种行动后，得到对应的奖励值即为状态-行动值函数。该行动依据策略得出，通过当前状态采用不同行动下环境反馈的奖励来确定。不同的策略会导致在相同状态下得到不同行动。在优化过程中，需要通过状态值函数和状态-行动值函数共同指导智能体的学习，最终达到较好直至最优的结果。

1.4　探索与开发

开发(exploitation)是在指定区域内进行最优值搜索，但是该最优值未必是全局最优，因此需要引入探索(exploration)对未知区域进行开发，以期望找到全局最优。图 1 所示的函数期望找到其全局最小点。假设当前局限在 $[a,0)$ 区间内进行开发，则开发的最终结果只能是在 x_1 处 $f(x)$ 取得最小值，而探索过程可以让搜索区间以一定概率拓展到 $[0,b]$ 上，从而有机会搜索到全局最优点 x_2。

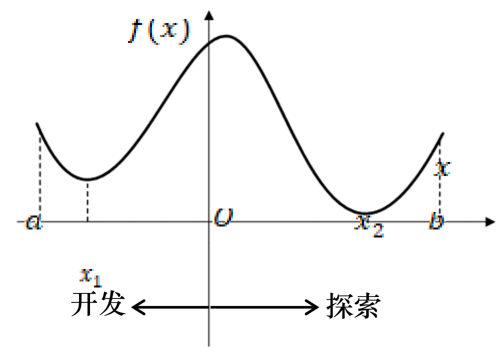


图 1　函数极小值的探索与开发过程

Fig. 1　Exploration and exploitation steps of searching for the minimum of a function.

开发过程是在区间内找到局部最优的过程，而探测则有机会跳出局部最优，从而找到全局最优的操作，实际应用表明，探索与开发相结合能够以较好的效果找到全局最优。

2　多目标航迹复原问题建模

图 2 所示是一般的强化学习的框架，智能体通过和环境需要进行一系列的交互，在每一个时刻，环境都处于某种状态，智能体根据当前的环境状态结合自己的历史行为准则及策略选择一定的行动 a，该行动会相应影响环境状态，环境也会根据智能体采取的行动作出相应响应，给出下一时刻的状态 s 即对当前所采取的行动的奖惩 r，重复该过程，直到环境不再发生响应和变化，即任务完成。

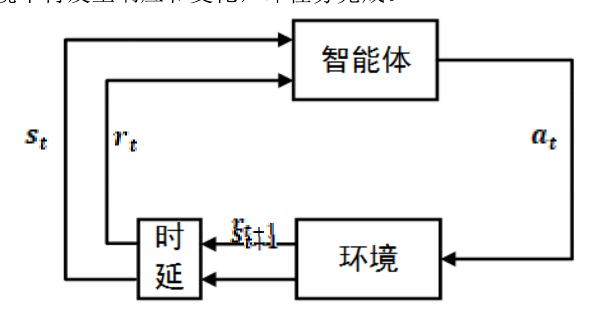


图 2　强化学习框架

Fig. 2　Framework of reinforcement learning

任务完成后，智能体进一步由环境最终的结局获得一个总分，该总分不同于每一步采取行动后获得的奖惩值，而是

任务完成后，根据任务完成的情况评判的总体得分，该得分是完成一次任务所对应的分数，因此又叫做单局得分。智能体的任务就是通过不断重复多轮次的完成任务，尝试找出利于单局得分高的操作，以希望得到最高的单局分数。

强化学习正是通过一系列不断的试错，逐步尝试应对环境各种状态时应该采取的措施并收集相应的奖励，最终获得成功。

2.1　多目标航迹复原问题及建模条件分析

探测器以一定频率探测若干目标点的地理位置，然而由于探测器本身的限制，无法确定每次探测到的各个点具体是哪一个目标，同时，会存在漏报和虚报的情况。例如，探测器探测到图 3 所示的信息。

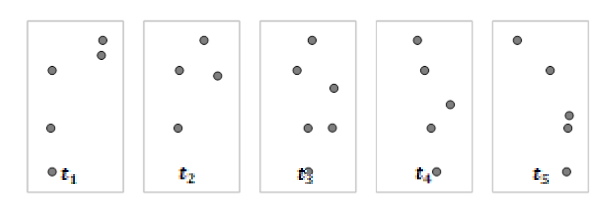


图 3　探测器在不同时刻探测到的目标地理位置示意图

Fig. 3　The sketch map of the multi target locations at each time

每个时刻 t_i 探测器对目标的位置进行探测，但是无法从探测结果上区分各个目标，例如图 3 中的 5 个时刻探测到的结果，并且在 t_2 时刻只探测到了 4 个目标，在 t_3 时刻探测到了 6 个目标，分别对应虚警和漏警。多目标航迹探测问题就是将这些探测的结果进行整合，得到各个目标的航迹，如图 4 所示。

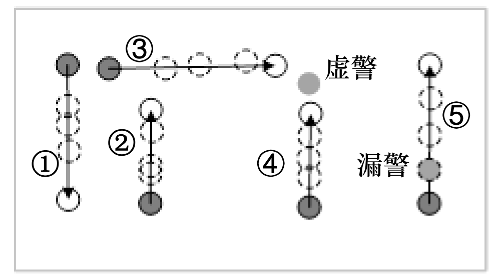


图 4　多目标航迹复原结果示意图

Fig. 4　Sketch map of the reconstruction trajectories with multi target

图 4 中带圆圈数字表示各个目标及其对应的航迹。箭头表示各个目标复原出航迹示意，箭尾实心圆圈表示目标的起始位置，空心圆圈表示目标终止位置，箭体虚线空心圆圈表示目标在路径上被探测器检测到的位置，灰色圆圈表示探测器虚警，灰色虚线圆圈表示探测器漏警。多目标航迹复原问题便是通过合理判断目标航迹总体的趋势，删除不合理的探测器虚警点，添加合理的漏警点，从而还原出目标的真实航迹。

实际场景中，一方面，由于探测器的局限性，只能探测到多个目标在各个时间点上的地理位置信息，即探测器无法提供每一帧数据中各个目标点对应具体是哪一个目标。另一方面，关注的目标具有一定的行动规律，不是纯粹的空间随机运动，即目标的速度具有一定的稳定性，不会出现过大的波动，目标一般会按照预先设定好的航迹进行移动，出现偏差时做微调而不会出现急转等现象。针对多目标航迹复原问题，在建模过程中主要考虑如下假设条件:

a)针对探测器探测条件的限制，主要有以下两个假设：(a)多个目标在每个检测时间点（即每一帧）上只被检测到地理位置信息，各目标之间无法区分；(b)每一帧中检测到的目标存在虚警或漏警的情况，即帧内存在噪声点。

b)针对目标运行具有一定规律性及合理性，主要考虑以下三点假设：(a)目标运行速度相对稳定，不会出现短期内突然增减速的情况；(b)目标运行轨迹相对光滑，不会出现急转弯等情况；(c)目标运行的轨迹限定在一定范围内的周期规律性运动，即目标完成的是在特定区域内的绕行任务。

实际目标在运行过程中可能会出现不符合上述假设的情况。一方面，某些目标航速存在波动，通过后续奖励函数的设计调整相应航距稳定项的正则化系数可以适应一定程度的航速波动，如果航速波动过大，只通过地理位置信息判断目标航距的误差也会相应较大;另一方面，由于探测器性能的影响，如果探测结果的每一帧中虚警的噪声点过多，也会错误地认为每一帧中存在的噪声点都是其他真实不存在的目标的航迹，该项的性能也可以通过减小目标个数项的正则化系数进行控制。

进一步分析航速稳定这一假设，实际情况中，目标航速确实会发生变化，本文考虑的场景中，目标会根据预先设定好的航迹进行运动，不存在急剧增减速的情况。因此，这一假设对应目标航速相对稳定,允许存在航速变化但变化不大。同时，针对航速尽量保持不变这一假设设计的奖励函数项，可以使还原出的航迹尽量在航速上稳定，避免相邻两帧中距离差距较远的目标点被划分为同一个目标的情况。在此假设下设计的奖励函数，还原出的目标航迹航速仍然可变，只是变化尽量不剧烈，比较符合实际目标的运行规律，即希望还原出的各个目标的航迹航速在合理范围内变化，而不会剧烈波动。

2.2　环境状态

系统所处的当前被探测到的目标位置视作当前系统的状态，在 t 时刻所处的状态记为 s_t，所有状态构成的空间记做 $\mathcal{S}$。如图 5 所示，图中的每个点 $p_i\,(i=1,2,3,4,5)$ 代表在不同时刻系统所处在的空间状态，然而实际接收的数据由于误差时间等限制，并不知道时间先后顺序，需要根据合理的推断选出一个合适的航迹点序。

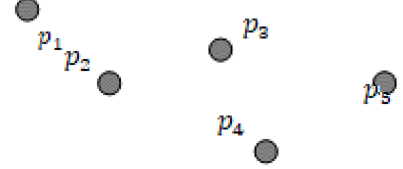


图 5　深度强化学习状态示意图

Fig. 5　Sketch map of the states in the deep reinforcement learing

2.3　行动

当系统处在环境状态 s_t 时，对 $t+1$ 时刻的行动空间是由前 t 个时刻状态决定的，即之前选过的状态不再参与行动备选，因此系统在 t 时刻的行动 $a_t \in \{s_i | s_i \in \mathcal{S}, i \neq 1,2,\ldots,t\}$，例如，在图 6 中 $t=2$ 时刻，假设已经选定了 $p_1 \rightarrow p_2$ 作为前两个时刻的航迹点，即 $s_1=p_1, s_2=p_2$，则在 $a_2=\{p_3,p_4,p_5\}$，即接下来的备选点只能从剩余未被选定为航迹点的探测点中选取。

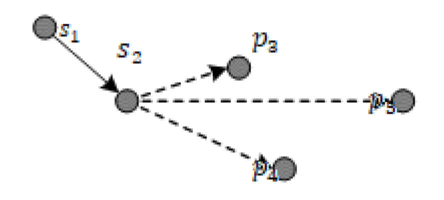


图 6　深度强化学习行动示意图

Fig. 6　Sketch map of the actions in the deep reinforcement learing

2.4　TOC 奖励函数

实际目标在目标中几乎不会改变航速，而对目标定位的检测点是在固定采样率条件下探测得到的，因此可以认为合理的航迹条件下，相邻两个点之间的距离应该相近，同时考虑目标运动过程中不一定沿直线运动，当曲线运动时，直连出的距离会比实际运动的曲线距离短(平面内任意连接两点的线中，直线段最短)，据此定义在 t 时刻系统状态为 s_t 的情况下采取行动 a_t 所得到的奖励为

$$\begin{aligned}&\text{reward}_t(a_t;s_t)\triangleq -d(s_t,a_t)-\\&\mathcal{D}(\{d(s_i,s_{i+1})\,|\,i=1,2,\ldots,t-1\}\cup\{d(s_t,a_t)\})\end{aligned}\tag{1}$$

这里 $\mathcal{D}(\circ)$ 表示样本集方差， $d(\circ,\circ)$ 表示两点之间的直线距离，也简记为 $d_{\circ\circ}$，该项称为航距稳定项。

进一步，若考虑目标飞行具有阶段性规律，例如某一段时间内会沿着一定曲线的航迹运动，下一段时间内可能会是另一个规律性较强的曲线，此时奖励可能不必要考虑所有之前状态之间距离的方差，而是只考虑一部分时间段内的方差，从而定义考虑前 n 个状态情况下的奖励函数为

$$\begin{aligned}&\text{reward}_n(a_t;s_t)\triangleq -d_{s_t,a_t}-\\&\mathcal{D}(\{d_{s_i,s_{i+1}}\,|\,i=t-n+1,\ldots,t-1\}\cup\{d_{s_t,a_t}\})\end{aligned}\tag{2}$$

这里 $n=1,2,\ldots,t$ ，且 $n=t$ 时对应为考虑所有状态情况下的奖励函数。

再次，考虑目标很少出现急转等现象，因此在选取行动策略上，如果选择了距离合适，但是却出现了急转等现象的航迹时，也应该相应获得负的奖励，采用曲率对该奖励进行刻画，选择曲率尽量小的航迹，相应的奖励函数增加曲率正则项如下:

$$\text{curv}_t(a_t;\text{s}_\text{t})\triangleq -\frac{\sqrt{4d_{s_t,a_t}^2 d_{s_{t-1},s_t}^2-\left(d_{s_t,a_t}^2+d_{s_{t-1},s_t}^2-d_{s_{t-1},a_t}^2\right)^2}}{d_{s_t,a_t}d_{s_{t-1},a_t}d_{s_{t-1},s_t}}\tag{3}$$

该式是利用相邻的三个采样点估计目标运行航迹的曲率的估计值，可以证明当采样间隔足够小，三个采样点足够近时该估计值趋近于目标运动轨迹的曲率，符合质点运动学中质点运动轨迹密切圆的定义，对该式的求解和说明见附录。

最后，考虑目标在移动时经常绕着较规律曲线进行移动，因此曲率的变化率前后也不应过大，对激励函数还需要加上一项历史曲率的方差作为正则项，类似航距稳定项，考虑前 n 个状态下曲率的方差，即为曲率稳定项:

$$\text{curvstable}_n(a_t;s_t)\triangleq -\mathcal{D}(\{\text{curv}_i|i=t-n+1,\cdots,t-1\})\tag{4}$$

综合以上各种结合实际目标轨迹的特点的因素，在 t 时刻，系统处于状态 s_t 时，采取行动 a_t 所对应的奖励函数为

$$r_t\cdot r(a_t;s_t\,|\,n)\triangleq \text{reward}_n(a_t;s_t)+\lambda_1\text{curv}_t(a_t;s_t)+\lambda_2\text{curvstable}_n(a_t;s_t)\tag{5}$$

这里考虑状态的阶数 $n=1,2,\ldots,t$ ，对应的 λ_1,λ_2 为正则项系数。在考虑多个目标时，需要进一步引入目标个数作为新增的正则项对奖励函数进行修正，只需要对上式再加上一个目标个数项作为第三个正则项即为本文提出的轨迹曲率圆(TOC)奖励函数。

2.5　Q 函数

奖励函数只是智能体在采取当前行动后，由环境反馈得到的奖励，智能体要学习的并不是当前一步环境反馈奖励最大，而是要经过序贯决策后最终结果最优，因此智能体的策略是基于特定状态 s 下，选择未来能够带来奖励最多的动作 a ，在特定状态 s 和行动 a 下，未来的奖励称为 Q 函数，表示为 $Q(s,a)$ 。

系统在 t 时刻的 Q 函数为 $Q(s_t,a_t)$ ，则在 t 时刻智能体的任务就是根据当前环境的状态 s_t 找出使 $Q(s_t,a)$ 最大的动作 a_t ，即

$$a_t=\underset{a}{argmax}\,Q(s_t,a|s_t)\tag{6}$$

然而，实际问题中的 Q 函数很难通过明确的显式表达写出，并且在实际应用中未必针对每一个状态和行为均有相同的 Q 函数，因此需要进一步通过迭代的方法逐步更新 Q 函数，迭代过程如下:

$$\begin{aligned}&Q_{k+1}(s_t,a_t)=Q_k(s_t,a_t)+\\&\alpha\left[r_{t+1}+\gamma\max_a Q_k(s_{t+1},a)-Q_k(s_t,a_t)\right]\end{aligned}\tag{7}$$

其中：α 表示学习率，γ 表示衰减因子。在深度强化学习中，该 Q 函数是由深度学习网络表示的，称为深度 Q 网络(deep Q network, DQN)[3]，输入数据是系统的状态 s ，输出是对应每个动作的 Q 值。同时，在深度强化学习中定义了一段记忆体，保存具体某一时刻的当前状态、奖励、动作、迁移到的下一个状态、状态是否结束等信息，定期从记忆体中随机选择固定大小的一段记忆，用于批量训练 Q 函数的深度神经网络。

3　算法流程

该算法的整体流程如图 7 所示。主要包含环境识别、DQN 以及采取行动三个过程。首先通过卷积神经网络(convolutional neural network, CNN)对环境进行状态识别和相应的奖励函数计算，将识别出的状态送入基于奖励函数计算的 Q 值训练过的 DQN 获得当前状态下取不同行动时的 Q 值，采取使 Q 值最大的行动作用于环境。

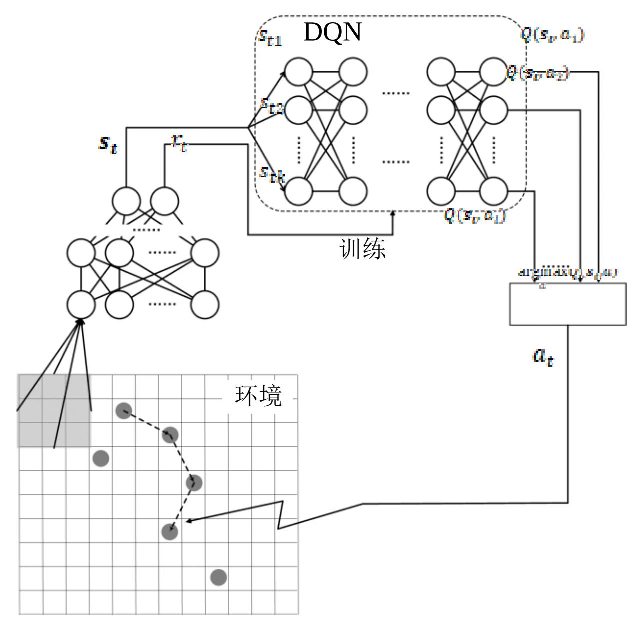


图 7　算法流程

Fig. 7　Flowchart of the algorithm

3.1　环境识别

对于深度强化学习的输入，本问题主要是获得了地理位置信息后的目标情况，需要考虑到对于一个图片，其应该具备平移旋转不变的特性。即目标平面的坐标原点及正交方向无论如何选取都不影响目标的航迹判别，应该采用 CNN 对其进行识别。环境识别采用的是 CNN，将获取到的目标在各个采样时刻的位置放入一定规格的表格网络中，用网络的整数坐标表示各个采样点所处的位置，具有目标点的网格以及各个目标的航速和方位角信息作为 CNN 的输入，同时将目标所处的状态作为样本标签对 CNN 进行训练，获得的网络可以作为智能体的视觉模块，为智能体提供环境当前的状态信息 s_t 以及采取一定行动后对应的奖励 r_t 。另一方面，由于

上一节中设计了该环境的奖励具有的函数形式，也可以直接搭建该函数对应的神经网络而不训练根据环境生成奖励的网络。

CNN 通过识别当前一帧图像中的数据网点输出该帧中的目标位置信息，CNN 的输入是每一帧图片，输出是检测到的每个目标的位置坐标。与复杂的目标检测问题不同的是，每一帧数据中目标不存在复杂的几何结构，而是探测到的一个个数据点。因此只需要将输入帧进行网格划分，并输出各个网格中是否存在目标点即可。CNN 的输入即为网格化的每一帧图片，同时具有等于网格个数的输出，每个输出用表示相应网格是否存在目标。

3.2 DQN

环境的状态包括连接到当前目标航迹的位置信息和航速、方位角等信息，是一个多维向量 $\boldsymbol{s}_t=[s_{t1},s_{t2},\ldots,s_{tk}]^{\mathrm{T}}$，即系统的状态为 k 维，将该向量作为 DQN 的输入，相应的 DQN 网络的输入也是 k 维度。DQN 的目的是根据环境所处的状态给出不同行动对应的 Q 函数值，因此需要通过奖励值递归计算出每个状态下对应各个行动的 Q 函数值对 DQN 进行训练，并且能够通过样本回放缓存区对这些数据再次访问以在需要时训练 DQN 网络。DQN 网络的输出是在当前状态下采取各个行动对应的 Q 函数值，即 $\{Q(\boldsymbol{s}_t,a)|a=a_1,a_2,\ldots,a_l\}$，这里的行动集即为获得的目标航迹点数 l，DQN 输出维度相应也为 l。

3.3 采取行动

在通过环境当前的状态计算出采取各个行动对应的 Q 函数值后，需要决定采取哪一行动来作用于环境，Q 函数值是综合考虑多步奖励回馈后对当前一步行动的估计，因此只需要取 Q 值对应最大的那个行动作用于环境即可。即采取的行动 a_t 满足

$$a_t=\arg\max_a Q(s_t,a) \tag{8}$$

环境相应改变后，环境识别 CNN 会进一步学习出改变后的环境状态和相应的奖励，如此循环往复，直到找出目标点的所有航迹信息。

4 实验分析

本文以 OpenAI Gym 为实验环境，采用该环境可以方便快捷地构建深度强化学习的环境，并构建 DQN 以对该环境进行强化学习。为了说明本文提出的 TOC 奖励函数的效果，分别将其在仿真航迹数据上实验 TOC 奖励函数中各个项对学习结果的影响，同时给出将该函数应用在实际数据集上的航迹复原效果。

4.1 实验设置

在以 OpenAI Gym 构造环境时，环境状态即为探测器探测到的目标地理位置，相应的行动也是各个位置，在计算回报函数时，如果选择的行动已经是构成轨迹的目标点，则相应策略回报为 0，否则按照本文给出的 TOC 回报函数结合已经被划分到轨迹内的目标点计算回报函数值。

TOC 回报函数中具有三个正则项，分别对应曲率大小、曲率稳定项及目标数目。为了给出 TOC 函数对深度强化学习用于多目标航迹挖掘问题的影响，本文的实验将 TOC 回报函数中的航距稳定项也作为正则项进行实验。当调整期中一个正则项系数时，其他正则项系数均取值为 1。实验用目标的真实航迹如图 8 所示。图中的横纵坐标分别是经纬度。

图 8 中共有四个目标，分别用四种颜色和形状区分，其中目标 1 和 2 分别进行直线运动，目标 3 和 4 分别进行圆周运动。轨迹上的每个点表示目标不同时刻下的准确地理位置。而实际上，探测器检测到的目标具有一定的损失和虚漏警，此处添加的随机噪声是均值为 0，方差为 5 的高斯噪声，同时虚警率漏警率均设置为 5%，如图 9 所示。

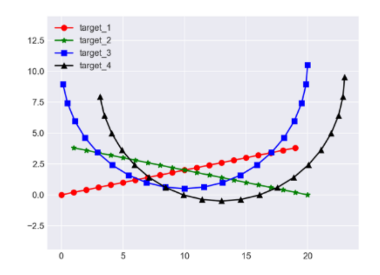


图 8 仿真出的目标真实轨迹

Fig. 8 Real trajectories of four targets in simulation

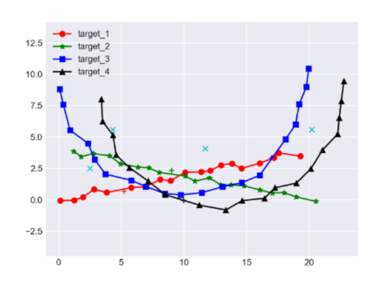


图 9 对仿真目标真实轨迹加噪

Fig 9 The noised real trajectories of four targets in simulation

图 9 中的轨迹是经过加噪处理后的仿真数据，并且每条轨迹都有用十字表示的漏警点，图中还有叉号表示的虚警点。由于漏警点在实验数据中并不存在，据此可以无须考虑此点，为了航迹复原的实验完全性，首先把虚漏警点考虑在内，实验结果如下。

4.2 航距稳定项

图 10 分别是航距稳定项的正则化系数取为 3 和 20 的还原效果图。

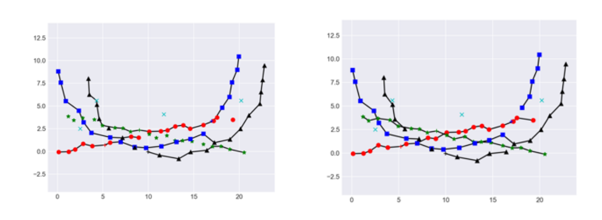


图 10 航距稳定项对航迹复原效果的影响(左右图分别为正则项系数为 3 和 20 时的航迹复原效果)

Fig. 10 Influence of the stability item of trajectory distance when the regular is 3 (the left figure) and 20 (the right figure)

从图 10 中可以看出，随着航距稳定项在奖励函数中的比重增加，复原出的目标航迹越来越趋于每一个目标的相邻两次探测目标点距离相同。从而目标数在该项正则项系数过大时出现错误。

4.3 曲率项

图 11 分别是有无曲率项，及随着曲率项的正则化系数增大的还原效果图。

曲率项是用来描述目标轨迹的曲率的，正常的目标在运行过程中很少出现急转弯的现象，因此该项的正则项系数越大表明控制学习出来的航迹越接近直线。从图 11 可以看出，当曲率项增大过程中，目标航迹逐渐变得越来越直，并且为

了构建直的航迹，牺牲目标个数这一项作为代价，从而导致目标个数的估计出现错误，右图中，大部分目标的轨迹都被还原成直线。

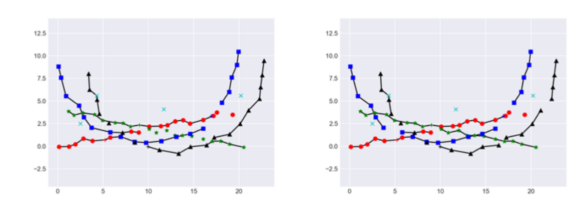


图 11　曲率项对航迹复原效果的影响

(左右图分别为正则项系数为 3 和 20 时的航迹复原效果)

Fig. 11　Influence of the curvature item when the regular is　3 (the left figure) and　20　(the right figure)

4.4　曲率稳定项

图 12 分别是有无曲率稳定项及随着曲率稳定项的正则化系数增大的还原效果图。

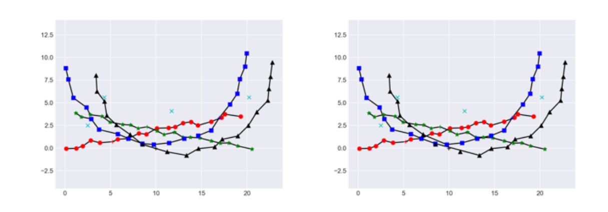


图 12　曲率稳定项对航迹复原效果的影响(左右图分别为正则项系数为 3 和 20 时的航迹复原效果)

Fig. 12　Influence of the stability item of trajectory curvature when the regular is　3　(the left figure) and　20　(the right figure).

曲率稳定项描述目标在运动过程中做圆周类运动时的现象，即目标虽然每一处都存在曲率，但是曲率保持圆周运动的曲率而不变。这对于具有规律性运动的目标具有较好的描述能力。从实验结果可以看出，随着该项系数的增加，目标的轨迹逐渐被判断为圆周运动，因为圆周的曲率是稳定不变的，从而该项为零，此时会牺牲目标个数作为代价。

4.5　目标个数项

图 13 分别是有无目标个数项，及随着目标个数项的正则化系数增大的还原效果图。

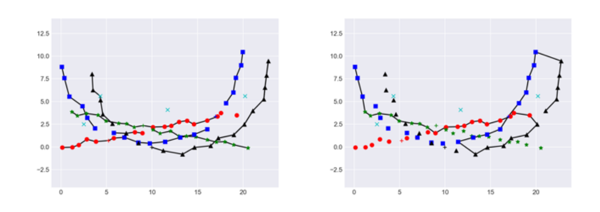


图 13　目标个数项对航迹复原效果的影响(左右图分别为正则项系数为 3 和 20 时的航迹复原效果)

Fig. 13　Influence of the stability item of the target number when the regular is　3　(the left figure) and　20　(the right figure).

目标个数一方面可以在知情的情况下给出，也可以由深度强化学习自动学习，然而该项的正则化系数需要仔细设置。如果已知目标个数，则尽量不试用该项。从实验结果可以看出，随着该项系数的增加，目标的航迹被区域连城一个目标的航迹，即期望目标个数尽量少。此时会将所有目标混在一起构造一条航迹。

4.6　与已有航迹聚类算法对比分析

本节以实际获取的目标地理位置数据集为实验对象，对本文提出的 TOC 奖励函数进行测试，综合考虑 TOC 奖励函数中的航距稳定、曲率、曲率稳定和目标个数项的影响，采用深度强化学习的实验结果如图 14 所示。

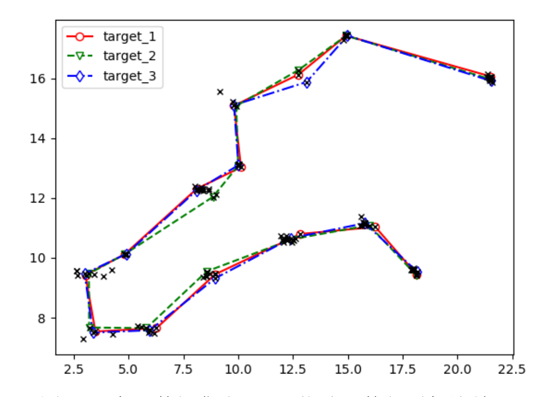


图 14　实际数据集上 TOC 奖励函数复原航迹结果

Fig. 14　Result of the trajectory reconstruction on the real dataset by the TOC reward function.

从图 14 中可以看出，还原出的目标数为 3 个，与实际目标数相符，同时采用 TOC 奖励函数后还原出的目标航迹在航距上分布稳定，否则会出现各个航迹点之间乱连，同一时刻内的多个噪声点相互连接的情况。航迹的曲率尽量小，总体上没有出现目标的轨迹特别尖锐的急转现象。曲率相对稳定，除了左下角的曲率较大外，其他处几乎相同。如果采用聚类的方法，由于聚类的信息此时只有距离信息，对于空间中的聚类结果，首先需要提前设置聚类个数为 3，即需要已知多目标航迹问题中的目标个数，系统无法自行学习出。然后根据各个点的二维地理位置信息进行聚类，采用航迹聚类算法[23]的复原结果如图 15 所示。

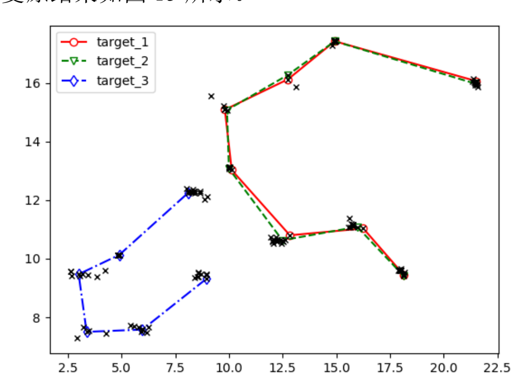


图 15　实际数据集采用航迹聚类方法结果

Fig. 15　The result of the trajectory reconstruction on the real dataset by the trajectory cluster method.

航迹聚类方法采用空间信息，根据空间距离进行聚类，该方法的聚类结果明显具有空间局限性，即主要考虑在欧式空间内一定距离内的目标航迹点形成轨迹。而基于 TOC 奖励函数的深度强化学习网络能够综合考虑航距、曲率、目标个数等多个指标综合给出轨迹复原结果。

另一方面，与已有的航迹聚类算法对比在目标个数判定方面本文提出的 TOC 奖励函数方法的性能。通过航迹聚类算法自动选定目标个数，可以通过分别计算不同目标个数情况下聚类结果选取，如图 16 所示。从图中可以看出，随着目标个数的增大，每个目标点被聚类为更多类别，从而每个类别内的聚类误差减小，一般情况下可以选定拐点作为聚类理想的类别数，因此该算法给出的目标个数为 2 个。而本文提出的 TOC 奖励函数下采用深度强化学习进行目标复原能够准确还原出与实际相符的 3 个目标点数。

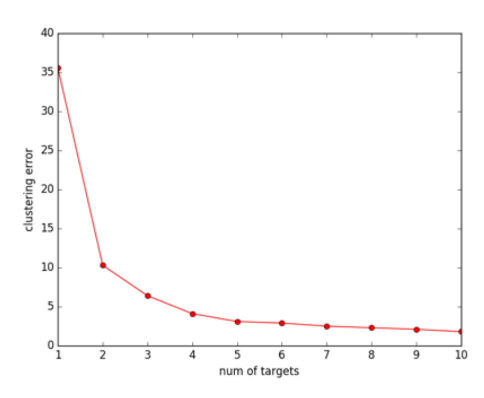


图 16　航迹聚类算法中目标个数与聚类误差之间的关系图

Fig. 16　Relationship between the number of targets and the clustering errors in the trajectory clustering algorithm.

5　结束语

实际应用问题中，检测器经常只能检测到目标的地理位置，而需要区分目标并将各个地理位置勾画成目标在这段时间内的航迹。本文根据目标航迹的物理意义构建数学模型，并提出了 TOC 奖励函数，同时给出该函数的数学证明。经过在仿真数据上对 TOC 奖励函数的各个项目进行学习效果对比，并在真实数据上进行实验后，证明了 TOC 奖励函数在衡量航距稳定、曲率、曲率稳定及目标数方面具有有效性。实验表明，通过调整 TOC 奖励函数中航距稳定项系数、曲率项系数、曲率稳定项系数及目标数项系数，能够有效控制深度强化学习的效果，复原出符合目标实际的航迹。需进一步解决的问题有：a)TOC 函数在指导深度强化学习航迹复原时，还存在训练不稳定的问题，有时不能较好地收敛到理想效果；b)进一步需要考虑更多实际目标航迹的物理信息，增加 TOC 函数的正则项，以从多种合理复原结果中找出最符合实际情况的目标航迹。由于探测器条件的限制，文中考虑的场景主要针对在仅已知地理位置信息的情况下引入其他先验信息，如航速航迹等特征信息，进行航距复原。提出的 TOC 奖励函数主要针对空间地理位置这一物理信息。在实际问题中，如果能够综合考虑更多物理信息，复原航迹将更加准确合理。目前可以考虑更多的物理信息包括：a)航向，这一信息需要依赖探测手段获取，实际中有较多探测手段可以获得这一信息，因此可以综合考虑航向信息并设计航向参数正则项加入到 TOC 奖励函数中；b)目标类型，这一信息需要其他探测手段获得，若已知目标类型，则可以进一步区分同一帧内的目标，从而合理关联已知类型目标的航迹，提高目标航迹复原准确性。实际问题中在获得目标类型这一类能够准确区分目标的物理信息后，可以通过该信息提前区分出能够识别出的目标，然后利用本文提出的方法处理剩余不能区分目标，从而进一步得到较好的航迹复原效果。

参考文献：

[1] Sutton R S, Barto A G. Reinforcement learning: an introduction [J]. IEEE Trans on Neural Networks, 1998, 9(5): 1054.

[2] Mnih V, Kavukcuoglu K, Silver D, *et al.* Playing atari with deep reinforcement learning [EB/OL]. (2013-01-01). https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf.

[3] Mnih V, Kavukcuoglu K, Silver D, *et al.* Human-level control through deep reinforcement learning [J]. Nature, 2015, **518** (7540): 529-533.

[4] Duan Yan, Chen Xi, Houthooft R, *et al.* Benchmarking deep reinforcement learning for continuous control [C]//Proc of International Conference on International Conference on Machine Learning. 2016: 1329-1338.

[5] Kai A, Deisenroth M P, Brundage M, *et al.* A Brief Survey of Deep Reinforcement Learning [J]. IEEE Signal Processing Magazine,2017, 8 (6) .

[6] Strehl A L, Li L, Wiewiora E, *et al.* PAC model-free reinforcement learning [C]//Proc of International Conference on Machine Learning. New York:ACM Press, 2006: 881-888.

[7] Lecun Y, Bengio Y, Hinton G. Deep learning [J]. Nature, 2015, 521 (7553): 436.

[8] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks [C]//Proc of International Conference on Artificial Intelligence and Statistics. 2011: 315-323.

[9] Li Yuxi. Deep reinforcement learning: an overview [EB/OL].2017. https://arxiv.org/abs/1701.07274

[10] Lampe T, Riedmiller M. Approximate model-assisted Neural Fitted Q-Iteration [C]//Proc of International Joint Conference on Neural Networks. Piscataway,NJ:IEEE Press, 2014.

[11] Schulman J, Levine S, Moritz P, *et al.* Trust region policy optimization [J]. Computer Science, 2015: 1889-1897.

[12] Hasselt H V, Guez A, Silver D. Deep reinforcement learning with double Q-learning [EB/OL]. 2015. https://arxiv.org/pdf/1509.06461.pdf

[13] Mnih V, Badia A P, Mirza M, *et al.* Asynchronous methods for deep reinforcement learning [C]// Proc of the 33rd International Conference on Machine Learning. 2016: 359-365.

[14] Schaul T, Quan J, Antonoglou I, *et al.* Prioritized experience replay [EB/OL]. 2015. https://arxiv.org/abs/1511.05952.

[15] Babaeizadeh M, Frosio I, Tyree S, *et al.* Reinforcement learning through asynchronous advantage actor-critic on a GPU [EB/OL]. https://openreview.net/pdf?id=r1VGvBcxl.

[16] Nair A, Srinivasan P, Blackwell S, *et al.* Massively PArallel methods for deep reinforcement learning [EB/OL]. 2015. https://arxiv.org/pdf/1507.04296.pdf.

[17] Silver D, Huang A, Maddison C J, *et al.* Mastering the game of Go with deep neural networks and tree search [J]. Nature, 2016, 529(7587): 484-489.

[18] Pilcer L S,Hoorelbeke A,Andigne A D. Playing flappy bird with deep reinforcement learning [C] , IEEE Trans on Neural Networks, 2015, 16 (1): 285-286.

[19] Qi Hang, Gong Jiang, Xu Lunbo. 3D flappy bird with reinforcement learning, 2016.

[20] Sallab A, Abdou M, Perot E, *et al.* Deep reinforcement learning framework for autonomous driving [J]. Electronic Imaging, 2017, 2017 (19): 70-76.

[21] Brockman G, Cheung V, Pettersson L, *et al.* OpenAI Gym [EB/OL]. https://gym.openai.com/.

[22] 王增福，潘泉，郎林，等．基于减法聚类的动态航迹聚类算法 [J]. 系统仿真学报,2009,21(16): 5240-5243,5246.(Wang Zengfu,Pan Quan,Lang Lin. Dynamic track cluster algorithm based on subtractive clustering [J]. Journal of System Simulation, 2009, 21(16): 5240-5243, 5246.)

[23] 陈勇．一种目标航迹数据聚类挖掘分析方法 [J]. 无线电工程，2015,45(3): 22-24. (Chen Yong. A data mining method for clustering target tracks [J] , Radio Engineering, 2015,45(3): 22-24.)

[24] 行艳妮，钱育蓉，南方哲，等. Spark 环境下 K-means 初始中心点优化研究综述 [J]. 计算机应用研究，2020，37(3) .

http://www.arocmag.com/article/02-2020-03-001.html (Xing Yanni, Qian Yurong, Nan Fangzhe, *et al.* Survey of optimization on K-means algorithm in Spark [J]. Application Research of Computers, 2020，37(3). http://www. arocmag. com/article/02-2020-03-001. html. .)

附录　TOC 奖励函数曲率稳定项推导

对于一条光滑曲线，其曲率的定义为

$$K \triangleq \left|\frac{\mathrm{d}\phi}{\mathrm{d}s}\right|,$$

这里的 ds 为曲线上固定点的弧长微元，即弧微分，$\mathrm{d}\phi$ 为切向角微元。当曲线以笛卡尔坐标表示为 $y=y(x)$ 的形式时，曲率表示为

$$K=\left|\frac{\mathrm{d}\phi}{\mathrm{d}s}\right|=\left|\frac{\mathrm{d}\arctan y'}{\sqrt{1+y'^2}\,\mathrm{d}x}\right|=\left|\frac{y''}{(1+y'^2)^{\frac{3}{2}}}\right|,$$

其中: $y'=\frac{\mathrm{d}y}{\mathrm{d}x}, y''=\frac{\mathrm{d}^2y}{\mathrm{d}x^2}$ 。

而在本文的问题中，只能获得曲线上的相隔较远的若干点，可以采用曲线上三点构成的外接圆半径倒数代替曲率，下面先给出给定三点后外接圆半径的求法，再对其极限与曲率半径之间的关系进行阐明。

如图 17 所示，对于给定的三个点，假设 A 点是已采取的策略 s_{t-1}，B 点是当前系统所处的状态 s_t，C 点是采取的行动 a_t，则相应的外接圆半径 R 满足正弦定理

$$\frac{|BC|}{\sin A}=\frac{|AC|}{\sin B}=\frac{|AB|}{\sin C}=2R \triangleq D,$$

简记为

$$\frac{a}{\sin A}=\frac{b}{\sin B}=\frac{c}{\sin C}=D,$$

从而

$$\begin{cases} a=D\sin A \\ b=D\sin B \\ c=D\sin(A+B) \end{cases},$$

$$c=D\sin(A+B)=D\sin A\cos B+D\cos A\sin B,$$

$$c=a\cos B+b\cos A,$$

$$c^2+b^2\cos^2 A-2bc\cos A=a^2\cos^2 B,$$

$$(c^2+b^2\cos^2 A-a^2\cos^2 B)^2=4b^2c^2\cos^2 A,$$

$$(c^2+b^2-a^2)^2=4b^2c^2\left(1-\frac{a^2}{D^2}\right),$$

$$1-\frac{a^2}{D^2}=\frac{(c^2+b^2-a^2)^2}{4b^2c^2},$$

$$\frac{1}{D}=\sqrt{\frac{4b^2c^2-(c^2+b^2-a^2)^2}{4a^2b^2c^2}},$$

$$\frac{1}{R}=\frac{\sqrt{4b^2c^2-(c^2+b^2-a^2)^2}}{abc}。$$

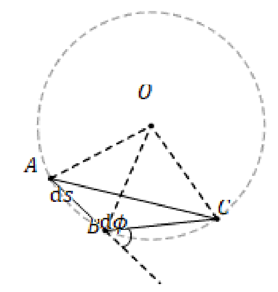


图 17　曲线微分示意图

Fig. 17　Sketch map the curve differential.

验证：

$$S_{\Delta ABC}=\frac{1}{2}ab\sin C=\frac{abc}{4R}=\frac{1}{4}\sqrt{4b^2c^2-(c^2+b^2-a^2)^2}=$$
$$\frac{1}{2}bc\sqrt{1-\frac{(c^2+b^2-a^2)^2}{4b^2c^2}}=\frac{1}{2}bc\sqrt{1-\cos^2 A}=\frac{1}{2}bc\sin A$$

由此结论可以写出外接圆半径的类似结论

$$\frac{1}{R}=\frac{\sqrt{4b^2c^2-(b^2+c^2-a^2)^2}}{abc}=$$
$$\frac{\sqrt{4a^2c^2-(a^2+c^2-b^2)^2}}{abc}=\frac{\sqrt{4a^2b^2-(a^2+b^2-c^2)^2}}{abc}$$

将 $a=d(s_t,a_t), b=d(s_{t-1},a_t), c=d(s_{t-1},s_t)$ 代入即得到曲率半径下的奖励函数的结论。下面，推导在对曲线采样情况下，外接圆半径可以近似等价于曲率半径。

$$a^2=\Delta t_2^2+[f(t+\Delta t_2)-f(t)]^2$$
$$b^2=(\Delta t_1+\Delta t_2)^2+[f(t+\Delta t_2)-f(t-\Delta t_1)]^2$$
$$c^2=\Delta t_1^2+[f(t-\Delta t_1)-f(t)]^2,$$

$$\lim_{\Delta t_2\to 0}\frac{a^2}{\Delta t_2^2}=1+f'^2(t)=\lim_{\Delta t_1\to 0}\frac{c^2}{\Delta t_1^2}=\lim_{\Delta t_1,\Delta t_2\to 0}\frac{b^2}{(\Delta t_1+\Delta t_2)^2},$$

$$\frac{\sqrt{4a^2c^2-(a^2+c^2-b^2)^2}}{abc}=\sqrt{\frac{4a^2c^2-(a^2+c^2-b^2)^2}{a^2b^2c^2}},$$

分母

$$\lim_{\Delta t_1,\Delta t_2\to 0}\frac{a^2b^2c^2}{\Delta t_1^2\Delta t_2^2(\Delta t_1+\Delta t_2)^2}=(1+f'^2(t))^3,$$

分子

$$\lim_{\Delta t_1,\Delta t_2\to 0}\frac{4a^2c^2-(a^2+c^2-b^2)^2}{\Delta t_1^2\Delta t_2^2(\Delta t_1+\Delta t_2)^2}。$$

对 $b^2=(\Delta t_1+\Delta t_2)^2+[f(t+\Delta t_2)-f(t-\Delta t_1)]^2$ 中第二项进行处理

$$f(t+\Delta t_2)-f(t-\Delta t_1)=f(t-\Delta t_1+\Delta t_1+\Delta t_2)-$$
$$f(t-\Delta t_1+\Delta t_2)+f(t-\Delta t_1+\Delta t_2)-f(t-\Delta t_1)=$$
$$\Delta t_1\cdot\frac{f(t-\Delta t_1+\Delta t_1+\Delta t_2)-f(t-\Delta t_1+\Delta t_2)}{\Delta t_1}+$$
$$f(t-\Delta t_1+\Delta t_2)-f(t-\Delta t_1)=$$
$$\Delta t_1\Delta t_2\cdot\left[\frac{f(t-\Delta t_1+\Delta t_1+\Delta t_2)-f(t-\Delta t_1+\Delta t_2)}{\Delta t_1}\right.$$

$$\left.-\frac{f(t)-f(t-\Delta t_1)}{\Delta t_1}\right]\Big/\Delta t_2$$

$$+\Delta t_1\frac{f(t)-f(t-\Delta t_1)}{\Delta t_1}$$

$$+\Delta t_2\frac{f(t-\Delta t_1+\Delta t_2)-f(t-\Delta t_1)}{\Delta t_2}。$$

在 $\Delta t_1,\Delta t_2\to 0$ 时

$$f(t+\Delta t_2)-f(t-\Delta t_1)\to f''(t)\Delta t_1\Delta t_2+f'(t)(\Delta t_1+\Delta t_2),$$

此时

$$b^2=(\Delta t_1+\Delta t_2)^2+[f(t+\Delta t_2)-f(t-\Delta t_1)]^2=$$
$$(1+f'^2(t))(\Delta t_1+\Delta t_2)^2+f''^2(t)\Delta t_1^2\Delta t_2^2+$$
$$2f'(t)f''(t)\Delta t_1\Delta t_2(\Delta t_1+\Delta t_2),$$

$$4a^2c^2-(a^2+c^2-b^2)^2=f''^2(t)\Delta t_1^2\Delta t_2^2(\Delta t_1+\Delta t_2)^2+$$
$$o\left(\Delta t_1^2,?\Delta t_2^2,(\Delta t_1+\Delta t_2)^2\right)$$

从而

$$\frac{4a^2c^2-(a^2+c^2-b^2)^2}{\Delta t_1^2\Delta t_2^2(\Delta t_1+\Delta t_2)^2}=f''^2(t)$$

即

$$\lim_{\Delta ABC}\frac{1}{R}=K。$$

另一方面，根据曲率的定义

$$K\triangleq\left|\frac{\mathrm{d}\phi}{\mathrm{d}s}\right|,$$

其倒数是曲线上该点密切圆的半径长，其中的微元 $\mathrm{d}\phi$ 在极限状态下

$$K=\left|\frac{\mathrm{d}\sin B}{\mathrm{d}s}\right|=\frac{\frac{\mathrm{d}b}{2R}}{\mathrm{d}s}=\frac{\frac{2\mathrm{d}s}{2R}}{\mathrm{d}s}=\frac{1}{R}$$

该结论与质点运动学中，质点运动轨迹上的密切圆是外接圆的极限之定义一致。